Root Refinement for Real Polynomials

Michael Kerber
Institute of Science and Technology (IST) Austria
Klosterneuburg, Austria
mkerber@ist.ac.at

Michael Sagraloff
Max-Planck-Institute for Informatics
Saarbrücken, Germany
msagralo@mpi-inf.mpg.de

Abstract

We consider the problem of approximating all real roots of a square-free polynomial f. Given isolating intervals, our algorithm refines each of them to a width of 2^{-L} or less, that is, each of the roots is approximated to L bits after the binary point. Our method provides a certified answer for arbitrary real polynomials, only considering finite approximations of the polynomial coefficients and choosing a suitable working precision adaptively. In this way, we get a correct algorithm that is simple to implement and practically efficient. Our algorithm uses the quadratic interval refinement method; we adapt that method to be able to cope with inaccuracies when evaluating f, without sacrificing its quadratic convergence behavior. We prove a bound on the bit complexity of our algorithm in terms of degree, size and separation of the roots, that is, parameters exclusively related to the geometric location of the roots. Our bound improves previous work on integer polynomials by a factor of $\deg f$ and essentially matches best known theoretical bounds on root approximation which are obtained by very sophisticated algorithms.

1 Introduction

The problem of computing the real roots of a polynomial in one variable is one of the best studied problems in mathematics. If one asks for a *certified* method that finds all roots, it is common to write the solutions as a set of disjoint *isolating* intervals, each containing exactly one root; for that reason, the term *real root isolation* is common in the literature. Simple, though efficient methods for this problem have been presented, for instance, based on Descartes' rule of signs [6], or on Sturm's theorem [7]. Recently, the focus of research shifted to polynomials with real coefficients which are approximated during the algorithm. It is worth to remark that this approach does not just generalize the integer case but has also lead to practical [10, 16] and theoretical [17] improvements of it.

We consider the related *real root refinement problem*: assuming that isolating intervals of a polynomial are known, *refine* them to a width of 2^{-L} or less (where $L \geq 0$ is an additional input parameter). Clearly, the combination of root isolation and root refinement, also called *strong root isolation*, yields a certified approximation of all roots of the polynomial to an absolute precision of 2^{-L} or, in other words, to L bits after the binary point in binary representation.

We present a solution to the root refinement problem for arbitrary square-free polynomials with real coefficients. Most of the related approaches are formulated in the REAL-RAM model where exact operations on real numbers are assumed to be available at unit costs. In contrast, our approach considers the coefficients as *bitstreams*, that is, it only works with finite prefixes of its binary representation, and we also quantify how many bits are needed in the worst case. The refinement uses the quadratic interval refinement method [1] (QIR for short) which is a quadratically converging hybrid of the bisection and secant method. We adapt the method to work with an increasing working precisions and use interval arithmetic to validate the correctness of the outcome. In this way, we obtain an algorithm that always returns a correct root approximation, is simple to implement on an actual computer (given that arbitrary approximations of the coefficients are accessible), and is adaptive in the sense that it might succeed with a much lower working precision than asserted by the worst-case bound.

We provide a bound on the bit complexity of our algorithm. To state it properly, we first define several magnitudes depending on the polynomial which remain fixed throughout the paper. Let

$$f(x) := \sum_{i=0}^{d} a_i x^i \in \mathbb{R}[x] \tag{1.1}$$

be a square-free polynomial of degree $d \geq 2$ with $|a_d| \geq 1$ and $\tau := \lceil \log(\max_i |a_i|) \rceil \geq 1$. We denote the roots of f by $z_1, \ldots, z_d$, and, w.l.o.g., we can assume that the roots are numbered such that the first m roots $z_1, \ldots, z_m$ are all the real roots of f. For each z_i, $\sigma_i = \sigma(z_i, f) := \min_{j \neq i} |z_i - z_j|$ denotes the *separation of* z_i, $\Sigma_f := \sum_{i=1}^{n} \log \sigma_i^{-1}$ and $\Gamma_f := \log(\max_i |z_i|)$ the *logarithmic root bound* of f. An interval $I = (a,b)$ is called *isolating* for a root z_i if I contains z_i and no other root of F. We set $\text{mid}(I) = \frac{a+b}{2}$ for the *center* and $w(I) := b - a$ for the *width* of I.

Main Result. *Given initial isolating intervals for the roots of f, our algorithm refines* one interval *to width* 2^{-L} *using*

$$\tilde{O}(d(d\Gamma_f + \Sigma_f)^2 + dL)$$

bit operations and refines all intervals *using*

$$\tilde{O}(d(d\Gamma_f + \Sigma_f)^2 + d^2 L)$$

bit operations, where $\tilde{O}$ *means that we ignore logarithmic factors. To do so, our algorithm requires the coefficients of f* in a precision *of at most*

$$\tilde{O}(d\Gamma_f + \Sigma_f + L)$$

bits after the binary point.

For the analysis, we divide the sequence of QIR steps in the refinement process into a *linear sequence* where the method behaves like bisection in the worst case, and a *quadratic sequence* where the interval is converging quadratically towards the root, following the approach in [11]. We do not require any conditions on the initial intervals except that they are disjoint and cover all real roots of F; an initial *normalization phase* modifies the intervals to guarantee the efficiency of our refinement strategy.

We remark that, using the recently presented root solver from [17], obtaining initial isolating intervals can be done with $\tilde{O}(d(d\Gamma_f + \Sigma_f)^2)$ bit operations using coefficient approximations of f to $\tilde{O}(d\Gamma_f + \Sigma_f)$ bits after the binary point. Combined with the latter result on root isolation, our complexity result also gives a bound on the strong root isolation problem.

The case of integer coefficients is often of special interest, and the problem has been investigated by previous work [11] for this restricted case. In the latter work, the complexity of root refinement was bounded by $\tilde{O}(d^4\tau^2 + d^3L)$. We improve this bound to

$$\tilde{O}(d^3\tau^2 + d^2L).$$

The difference in the complexities is due to a different approach to evaluate the sign of f at rational points which is the main operation in the refinement procedure: for an interval of size $2^{-\ell}$, the evaluation of f at the endpoints of the interval has a complexity of $\tilde{O}(d^2(\tau+\ell))$ when using exact rational arithmetic because evaluated function values can consist of up to $d(\tau+\ell)$ bits. However, we show that we can still compute the sign of the function value with certified numerical methods using the substantially smaller working precision of $O(d\tau+\ell)$. We remark that the latter result certainly only applies to points whose distance to a root is not much smaller than $2^{-\ell}$, thus, we modified the QIR method in way such that the latter requirement is given.

Related work. The problem of accurate root approximation is omnipresent in mathematical applications; certified methods are of particular importance in the context of computations with algebraic objects, for instance, when computing the topology of algebraic curves [5, 9] or when solving systems of multivariate equations [2].

The idea of combining bisection with a faster converging method to find roots of continuous functions has been first introduced in *Dekker's method* and elaborated in *Brent's method*; see [4] for a summary. However, these approaches assume exact arithmetic for their convergence results.

For polynomial equations, numerous algorithms are available, for instance, the *Jenkins-Traub algorithm* or *Durant-Kerner iteration*; although they usually approximate the real roots very fast in practice, general worst-case bounds on their arithmetic complexity are not available. In fact, for some variants, even termination cannot be guaranteed in theory; we refer to the survey [15] for extensive references on these and further methods.

The theoretical complexity of root approximation has been investigated by Pan [14]. Assuming all roots to be in the unit disc, he achieves a bit complexity of $\tilde{O}(n^3 + n^2L)$ for approximating all roots to an accuracy of 2^{-L}, which matches our bound if L is the dominant input parameter. His approach even works for polynomials with multiple roots. However, as Pan admits in [15], the algorithm is difficult to implement and so is the complexity analysis when taking rounding errors in intermediate steps into account. Moreover, it appears unclear whether his bound can be improved if only a single root needs to be approximated.

We improve on the first version of this paper [12] in two ways: first of all, in our bit complexity result, we remove the dependence on the coefficient size and, thus, relate the hardness of root approximation to parameters that exclusively depend on the geometric location of the roots; we shortly expose in Section 6 who to benefit from this approach. Also, in this work, we redefine the threshold for the interval width that guar-

Algorithm 1 EQIR: Exact Quadratic Interval Refinement

```
INPUT: $f \in \mathbb{R}[x]$ square-free, $I = (a,b)$ isolating, $N = 2^{2^i} \in \mathbb{N}$
OUTPUT: $(J, N')$ with $J \subseteq I$ isolating for $\xi$ and $N' \in \mathbb{N}$
procedure EQIR($f, I = (a,b), N$)
    if $N = 2$, return (BISECTION($f$,$I$),4).
    $\omega \leftarrow \frac{b-a}{N}$
    $m' \leftarrow a + \mathrm{round}(N\frac{f(a)}{f(a)-f(b)})\omega$        $\triangleright\ m' \approx a + \frac{f(a)}{f(a)-f(b)}(b-a)$
    $s \leftarrow \mathrm{sign}(f(m'))$
    if $s = 0$, return $([m', m'], \infty)$
    if $s = \mathrm{sign}(f(a))$ and $\mathrm{sign}(f(m'+\omega)) = \mathrm{sign}(f(b))$, return $([m', m'+\omega], N^2)$
    if $s = \mathrm{sign}(f(b))$ and $\mathrm{sign}(f(m'-\omega)) = \mathrm{sign}(f(a))$, return $([m'-\omega, m'], N^2)$
    Otherwise, return $(I, \sqrt{N})$.
end procedure
```

antees quadratic convergence (Defintion 12); in this way, we get rid of the magnitude $R = \log|\mathrm{res}(f, f')|^{-1}$, which is a pure artifact of the analysis of [12].

Outline. We summarize the (exact) QIR method in Section 2. A variant using only approximate coefficients is described in Section 3. Its precision demand is analyzed in Section 4. Based on that analysis of a single refinement step, the complexity bound of root refinement is derived in Section 5. We end with concluding remarks in Section 6.

2 Review on exact QIR

Abbott's QIR method [1, 11] is a hybrid of the simple (but inefficient) bisection method with a quadratically converging variant of the *secant method.* We refer to this method as EQIR, where "E" stands for "exact" in order to distinguish from the variant presented in Section 3. Given an isolating interval $I = (a,b)$ for a real root ξ of f, we consider the secant through $(a, f(a))$ and $(b, f(b))$ (see also Figure 3.1). This secant intersects the real axis in the interval I, say at x-coordinate m. For I small enough, the secant should approximate the graph of the function above I quite well and, so, $m \approx \xi$ should hold. An EQIR step tries to exploit this fact:

The isolating interval I is (conceptually) subdivided into N subintervals of same size, using $N+1$ equidistant grid points. Each subinterval has width $\omega := \frac{w(I)}{N}$. Then m', the closest grid point to m, is computed and the sign of $f(m')$ is evaluated. If that sign equals the sign of $f(a)$, the sign of $f(m'+\omega)$ is evaluated. Otherwise, $f(m'-\omega)$ is evaluated. If the sign changes between the two computed values, the interval $(m', m'+\omega)$ or the interval $(m'-\omega, m')$, respectively, is set as new isolating interval for ξ. In this case, the EQIR step is called *successful*. Otherwise, the isolating interval remains unchanged, and the EQIR step is called failing. See Algorithm 1 for a description in pseudo-code.

In [11], the root refinement problem is analyzed using the just described EQIR method for the case of integer coefficients and exact arithmetic with rational numbers.

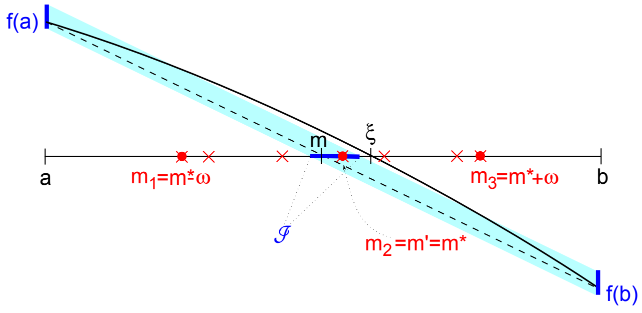

Figure 3.1: Illustration of an AQIR step for $\boldsymbol{N = 4}$.

For that, a sequence of EQIR steps is performed with $N = 4$ initially. After a successful EQIR step, N is squared for the next step; after a failing step, N is set to $\sqrt{N}$. If N drops to 2, a bisection step is performed, and N is set to 4 for the next step. In [11], a bound on the size of an interval is provided to guarantee success of every EQIR and, thus, quadratic convergence of the overall method.

3 Approximate QIR

The most important numerical operation in an EQIR step is the computation of $f(x_0)$ for values $x_0 \in I$. Note that $f(x_0)$ is needed for determining the closest grid point m' to the secant (Step 4 of Algorithm 1), and its sign is required for checking for sign changes in subintervals (Steps 6-8).

What are the problems if f is a bitstream polynomial as in (1.1), so that $f(x_0)$ can only be evaluated up to a certain precision? First of all, $\frac{Nf(a)}{f(a)-f(b)}$ can only be computed approximately, too, which might lead to checking the wrong subinterval in the algorithm if m is close to the center of a subinterval. Even more seriously, if $f(x_0)$ is zero, then, in general, its sign can never be evaluated using any precision. Even if we exclude this case, the evaluation of $f(x_0)$ can become costly if x_0 is too close to a root of f. The challenge is to modify the QIR method such that it can cope with the uncertainties in the evaluation of f, requires as few precision as possible in a refinement step and still shows a quadratic convergence behavior eventually.

Bisection is a subroutine called in the QIR method if $N = 2$; before we discuss the general case, we first describe our variant of the bisection in the bitstream context. Note that we face the same problem: Writing $\mathrm{mid}(I)$ as the center of $I = (a,b)$, $f(\mathrm{mid}(I))$ might be equal or almost equal to zero. We will overcome this problem by evaluating f at several x-coordinates "in parallel". For that, we subdivide I into 4 equally wide parts using the subdivision points $m_j := a + j \cdot \frac{b-a}{4}$ for $1 \le j \le 3$. We also assume that the sign of f at a is already known. We choose a starting precision ρ and compute $f(m_1), \ldots, f(m_3)$ using interval arithmetic in precision ρ (cf. Section 4 for details). If less than 2 out of 3 signs have been determined using precision ρ, we set $\rho \leftarrow 2\rho$ and

Algorithm 2 Approximate Bisection

INPUT: $f \in \mathbb{R}[x]$ square-free, $I = (a,b)$ isolating, $s = \operatorname{sign}(f(a))$
OUTPUT: $J \subseteq I$ isolating with $2 \cdot w(J) \leq w(I)$.

```
procedure APPROXIMATE_BISECTION($f, I = (a,b), s$)
    $V \leftarrow [a + (i-1) \cdot \frac{b-a}{4}, i = 1, \ldots, 5]$
    $S = [s, 0, 0, 0, -s]$
    $\rho \leftarrow 2$
    while $S$ contains more than one zero do
        for i=2,...,4 do
            If $S[i] = 0$, set $S[i] \leftarrow \operatorname{sign}\mathfrak{B}(f(V[i]), \rho)$
        end for
        $\rho \leftarrow 2\rho$
    end while
    Find $v, w$, such that $S[v] \cdot S[w] = -1 \wedge (v+1 = w \vee (v+2 = w \wedge S[v+1] = 0))$
    return $(V[v], V[w])$
end procedure
```

repeat the calculation with increased precision. Once the sign at at least 2 subdivision points is determined, we can determine a subinterval of at most half the size of I that contains ξ (Algorithm 2). We will refer to this algorithm as “bisection”, although the resulting interval can also be only a quarter of the original size. Note that f can only become zero at one of the subdivision points which guarantees termination also in the bitstream context. Moreover, at least 2 of the 3 subdivision points have a distance of at least $\frac{b-a}{8}$ to ξ. This asserts that the function value at these subdivision points is reasonable large and leads to an upper bound of the required precision (Lemma 5).

We next describe our bitstream variant of the QIR method that we call *approximate quadratic interval refinement*, or AQIR for short (see also Figure 3.1 for the illustration of an AQIR step for $N = 4$). Compared to the exact variant, we replace two substeps. In Step 4, we replace the computation of $\lambda := N\frac{f(a)}{f(a)-f(b)}$ as follows: For a working precision ρ, we evaluate $f(a)$ and $f(b)$ via interval arithmetic with precision ρ (blue vertical intervals in the above figure) and evaluate $N\frac{f(a)}{f(a)-f(b)}$ with interval arithmetic accordingly (cf. Section 4). Let $J = (c,d)$ denote the resulting interval (in Figure 3.1, $\mathscr{I} = a + J \cdot \frac{b-a}{N}$ is the intersection of the stripe defined by the interval evaluations of $f(a)$ and $f(b)$ with the real axis). If the width $w(J)$ of J is more than $\frac{1}{4}$, we set ρ to 2ρ and retry. Otherwise, let ℓ be the integer closest to $\operatorname{mid}(J)$ and set $m^* := a + \ell \cdot \frac{b-a}{N}$. For $m = a + \frac{f(a)}{f(a)-f(b)}(b-a)$ as before and $m_j := a + j \cdot \frac{b-a}{N}$ (red dots) for $j = 0, \ldots, N$, the following Lemma shows that the computed $m^* = m_\ell$ indeed approximates m on the m_j-grid:

Lemma 1. *Let m be inside the subinterval $[m_j, m_{j+1}]$. Then, $m^* = m_j$ or $m^* = m_{j+1}$. Moreover, let $m' \in \{m_j, m_{j+1}\}$ be the point that is closer to m. If $|m - m'| < \frac{b-a}{4N}$, then $m^* = m'$.*

Proof. Let $\lambda := N\frac{f(a)}{f(a)-f(b)}$ and J the interval computed by interval arithmetic as above, with width at most $\frac{1}{4}$. Since $m = f(a) + \lambda\frac{b-a}{N} \in [m_j, m_{j+1}]$, it follows that $j \leq \lambda \leq j+1$. By construction, $\lambda \in J$. Therefore, $|\lambda - \mathrm{mid}(J)| \leq \frac{1}{8}$ and, thus, it follows that $\mathrm{mid}(J)$ can only be rounded to j or $j+1$. Furthermore, for $m' = m_j$, $|m - m'| < \frac{b-a}{4N}$ implies that $|\lambda - j| < \frac{1}{4}$. It follows that $|\mathrm{mid}(J) - j| < \frac{3}{8}$ by triangle inequality, so $\mathrm{mid}(J)$ must be rounded to j. The case $m' = m_{j+1}$ is analogous. □

The second substep to replace in the QIR method is to check for sign changes in subintervals in Steps 6-8. As before, we set $\omega := w(I)/N$. Instead of comparing the signs at m' and $m' \pm \omega$, we choose the seven subdivision points (red crosses in Figure 3.1)

$$m^* - \omega, m^* - \frac{7\omega}{8}, m^* - \frac{\omega}{2}, m^*, m^* + \frac{\omega}{2}, m^* - \frac{7\omega}{8}, m^* + \omega. \tag{3.1}$$

In case that $m^* = a$ or $m^* = b$, we only choose the 4 points of (3.1) that lie in I. For a working precision ρ, we evaluate the sign of f at all subdivision points using interval arithmetic. If the sign remains unknown for more than one point, we set ρ to 2ρ and retry. After the sign is determined for all except one of the points, we look for a sign change in the sequence. If such a sign change occurs, we set the corresponding interval I^* as isolating and call the AQIR step *successful*. Otherwise, we call the step *failing* and keep the old isolating interval. As in the exact case, we square up N after a successful step, and reduce it to its square root after a failing step. See Algorithm 3 for a complete description.

Note that, in case of a successful step, the new isolating interval I^* satisfies $\frac{1}{8N}w(I) \leq w(I^*) \leq \frac{1}{N}w(I)$. Also, similar to the bisection method, the function can only be zero at one of the chosen subdivision points, and the function is guaranteed to be reasonably large for all but one of them, which leads to a bound on the necessary precision (Lemma 7). The reader might wonder why we have chosen a non-equidistant grid involving the subdivision points $m^* \pm \frac{7}{8}\omega$. The reason is that these additional points allow us to give a success guarantee of the method under certain assumptions in the following lemma which is the basis to prove quadratic convergence if the interval is smaller than a certain threshold (Section 5.2).

Lemma 2. *Let $I = (a,b)$ be an isolating interval for some root ξ of f, $s = \mathrm{sign}(f(a))$ and m as before. If $|m - \xi| < \frac{b-a}{8N} = \frac{\omega}{8}$, then* AQIR*$(f,I,N,s)$ succeeds.*

Proof. Let m^* be the subdivision point selected by the AQIR method. We assume that $m^* \notin \{a,b\}$; otherwise, a similar (simplified) argument applies. By Lemma 1 $m \in [m^* - \frac{3}{4}\omega, m^* + \frac{3}{4}\omega]$ and, thus, $\xi \in (m^* - \frac{7}{8}\omega, m^* + \frac{7}{8}\omega)$. It follows that the leftmost two points of (3.1) have a different sign than the rightmost two points of (3.1). Since the sign of f is evaluated for at least one value on each side, the algorithm detects a sign change and, thus, succeeds. □

4 Analysis of an AQIR step

The running time of an AQIR step depends on the maximal precision ρ needed in the two while loops (Step 5, Steps 11-15) of Algorithm 3. The termination criterion of both loops is controlled by evaluations of the form $\mathfrak{B}(E,\rho)$, where E is some polynomial expression and ρ is the current working precision.

We specify recursively what we understand by evaluating E in precision ρ with interval arithmetic. For that, we define $\mathrm{down}(x,\rho)$ for $x \in \mathbb{R}$ and $\rho \in \mathbb{N}$ to be the maximal $x_0 \leq x$ such that $x_0 = \frac{k}{2^\rho}$ for some integer k. The same way $\mathrm{up}(x,\rho)$ is the minimal $x_0 \geq x$ with x_0 of the same form. We extend this definition to arithmetic expressions by the following rules (we leave out ρ for brevity):

$$\begin{aligned}
\mathrm{down}(E_1+E_2) &:= \mathrm{down}(E_1)+\mathrm{down}(E_2)\\
\mathrm{up}(E_1+E_2) &:= \mathrm{up}(E_1)+\mathrm{up}(E_2)\\
\mathrm{down}(E_1\cdot E_2) &:= \mathrm{down}(\min\{\mathrm{down}(E_1)\mathrm{down}(E_2),\mathrm{up}(E_1)\mathrm{up}(E_2),\\
&\qquad \mathrm{up}(E_1)\mathrm{down}(E_2),\mathrm{down}(E_1)\mathrm{up}(E_2)\})\\
\mathrm{up}(E_1\cdot E_2) &:= \mathrm{up}(\max\{\mathrm{down}(E_1)\mathrm{down}(E_2),\mathrm{down}(E_1)\mathrm{up}(E_2),\\
&\qquad \mathrm{up}(E_1)\mathrm{down}(E_2),\mathrm{up}(E_1)\mathrm{up}(E_2)\})\\
\mathrm{down}(1/E_1) &:= \mathrm{down}(1/\mathrm{up}(E_1))\\
\mathrm{up}(1/E_1) &:= \mathrm{up}(1/\mathrm{down}(E_1))
\end{aligned}$$

Finally, we define the interval $\mathfrak{B}(E,\rho) := [\mathrm{down}(E,\rho),\mathrm{up}(E,\rho)]$. By definition, the exact value of E is guaranteed to be contained in $\mathfrak{B}(E,\rho)$. We assume that polynomials $f \in \mathbb{R}[x]$ are evaluated according to the Horner scheme, and when evaluating $f(c)$ with precision ρ, the above rules apply in each arithmetic step. The next lemma provides a worst case bound on the size of the resulting interval $\mathfrak{B}(f(c),\rho)$ under certain conditions. We further remark that, in an actual implementation, $\mathfrak{B}(E,\rho)$ is usually much smaller than the worst case bound derived here. Nevertheless, our complexity analysis is based on the latter bound. Throughout the following considerations , $\Gamma \in \mathbb{N}$ denotes an integer upper bound on the root bound Γ_f, that is, $\Gamma \geq \Gamma_f$, and, in particular $\log|z_i| \leq \Gamma$ for all roots z_i of f.

Lemma 3. *Let f be a polynomial as in (1.1), $c \in \mathbb{R}$ with $|c| \leq 2^{\Gamma+2}$, and $\rho \in \mathbb{N}$. Then,*

$$|f(c)-\mathrm{down}(f(c),\rho)| \leq 2^{-\rho+1}(d+1)^2 2^{\tau+d(\Gamma+2)} \tag{4.1}$$

$$|f(c)-\mathrm{up}(f(c),\rho)| \leq 2^{-\rho+1}(d+1)^2 2^{\tau+d(\Gamma+2)} \tag{4.2}$$

In particular, $\mathfrak{B}(f(c),\rho)$ has a width of at most $2^{-\rho+2}(d+1)^2 2^{\tau+d(\Gamma+2)}$.

Proof. We do induction on d. The statement is clearly true for $d=0$. For $d>0$, we write $f(c) = a_0 + cg(c)$ with $a_0 \in \mathbb{R}$ the constant coefficient of f and g of degree $d-1$. Note that, for any real value x, $|\mathrm{down}(x,q)-x| < 2^{-\rho}$, same for up. Therefore, we can bound as follows (again, leaving ρ out for simplicity):

$$\begin{aligned}
|f(c)-\mathrm{down}(f(c))| &= |a_0+cg(c)-\mathrm{down}(a_0+cg(c))| = |a_0+cg(c)-\mathrm{down}(a_0)-\mathrm{down}(cg(c))|\\
&\leq |cg(c)-\mathrm{down}(cg(c))|+2^{-\rho}
\end{aligned}$$

Note that $\mathrm{down}(c\cdot g(c)) = \mathrm{down}(H_1(c)\cdot H_2(g(c)))$ where $H_{1,2} = \mathrm{down}$ or $H_{1,2} = \mathrm{up}$.

Algorithm 3 Approximate Quadratic interval refinement

```
INPUT: $f \in \mathbb{R}[x]$ square-free, $I = (a,b)$ isolating, $N = 2^{2^i} \in \mathbb{N}$, $s = \mathrm{sign}(f(a))$
OUTPUT: $(J,N')$ with $J \subseteq I$ isolating and $N' \in \mathbb{N}$
procedure AQIR($f, I = (a,b), N$)
    if $N = 2$, return (APPROXIMATE_BISECTION($f,I,s$),4).
    $\omega \leftarrow \frac{b-a}{N}$
    $\rho \leftarrow 2$
    while $J \leftarrow \mathfrak{B}(N\frac{f(a)}{f(a)-f(b)}, \rho)$ has width $> \frac{1}{4}$, set $\rho \leftarrow 2\rho$
    $m^* \leftarrow a + \mathrm{round}(\mathrm{mid}(J)) \cdot \omega$
    if $m^* = a$, $s \leftarrow 4, V \leftarrow [m^*, m^* + \frac{1}{2}\omega, m^* + \frac{7}{8}\omega, m^* + \omega], S \leftarrow [s,0,0,0]$
    if $m^* = b$, $s \leftarrow 4, V \leftarrow [m^* - \omega, m^* - \frac{7}{8}\omega, m^* - \frac{1}{2}\omega, m^*], S \leftarrow [0,0,0,-s]$
    if $a < m^* < b$, $s \leftarrow 7, V \leftarrow [m^* - \omega, m^* - \frac{7}{8}\omega, m^* - \frac{1}{2}\omega, m^*, m^* + \frac{1}{2}\omega, m^* + \frac{7}{8}\omega, m^* + \omega], S \leftarrow [0,0,0,0,0,0,0]$
    $\rho \leftarrow 2$
    while $S$ contains more than one zero do
        for i=1,...,s do
            If $S[i] = 0$, set $S[i] \leftarrow \mathrm{sign}\,\mathfrak{B}(f(V[i]), \rho)$
        end for
        $\rho \leftarrow 2\rho$
    end while
    If $\exists v,w : S[v] \cdot S[w] = -1 \wedge (v+1 = w \vee (v+2 = w \wedge S[v+1] = 0))$ return $((V[v],V[w]),N^2)$
    Otherwise, return $(I, \sqrt{N})$
end procedure
```

Moreover, we can write $H_1(c) = c - \varepsilon$ with $|\varepsilon| < 2^{-\rho}$. Therefore, we can rearrange

$$\begin{aligned} & |cg(c) - \mathrm{down}(cg(c))| + 2^{-\rho} \leq |cg(c) - (c - \varepsilon) \cdot H_2(g(c))| + 2^{-\rho+1} \\ \leq \quad & |cg(c) - c \cdot H_2(g(c))| + |\varepsilon| \cdot |H_2(g(c))| + 2^{-\rho+1} \\ \leq \quad & |c| \cdot |g(c) - H_2(g(c))| + 2^{-\rho}|H_2(g(c))| + 2^{-\rho+1} \end{aligned}$$

By a simple inductive proof on the degree, we can show that both $|\mathrm{up}(g(c))|$ and $|\mathrm{down}(g(c))|$ are bounded by $d2^{\tau+d(\Gamma+2)}$. Using that and the induction hypothesis yields

$$\begin{aligned} & |c| \cdot |g(c) - h(g(c))| + 2^{-\rho}|H_2(g(c))| + 2^{-\rho+1} \\ < \quad & 2^{\Gamma+2}2^{-\rho+1}d^2 2^{\tau+(d-1)(\Gamma+2)} + 2^{-\rho}d2^{\tau+d(\Gamma+2)} + 2^{-\rho+1} \\ \leq \quad & 2^{-\rho+1}(d^2+d+1)2^{\tau+d(\Gamma+2)} \leq 2^{-\rho+1}(d+1)^2 2^{\tau d} \end{aligned}$$

The bound for $|f(c) - \mathrm{up}(f(c))|$ follows in the same way. □

For the sake of simplicity, we decided to assume fixed-point arithmetic, that means, ρ determines the number of bits *after the binary point.* We refer the interested reader to [13, Thm. 12], where a corresponding result for floating-point arithmetic is given.

We analyze the required working precision of approximate bisection and of an AQIR step next. We exploit that, whenever we evaluate f at t subdivision points, $t-1$ of them have a certain minimal distance to the root in the isolating interval. The following lemma gives a lower bound on $|f(x_0)|$ for such a point x_0, given that it is sufficiently far away from any other root of f.

Lemma 4. *Let f be as in (1.1), $\xi = z_{i_0}$ a real root of f and x_0 be a real value with distance $|x_0 - z_i| \geq \frac{\sigma_i}{4}$ to all* real *roots $z_i \neq z_{i_0}$. Then,*

$$|f(x_0)| > |\xi - x_0| \cdot 2^{-(2d+\Gamma+\Sigma_f)}.$$

(recall the notations from Section 1 for the definitions of σ_i and Σ_f)

Proof. For each non-real root z_i of f, there exists a complex conjugate root $\bar{z}_i$ and, thus, we have $|x_0 - z_i| \geq \operatorname{Im}(z_i) \geq \frac{\sigma_i}{2} > \frac{\sigma_i}{4}$ for all $i = m+1, \ldots, d$ as well. It follows that

$$\begin{aligned}|f(x_0)| &= |a_d \prod_{i=1}^{d}(x_0 - z_i)| = |a_d| \cdot |\xi - x_0| \cdot \prod_{i=1,\ldots,d:i\neq i_0} |x_0 - z_i| \\ &\geq |\xi - x_0| \cdot \frac{4}{\sigma_{i_0}} \cdot \prod_{i=1}^{d} \frac{\sigma_i}{4} > |\xi - x_0| \cdot 2^{-2d-\Gamma} \cdot 2^{-\Sigma_f},\end{aligned}$$

where the last inequality uses that $|z_i| \leq 2^{\Gamma}$ and, thus, $\sigma(z_i) \leq 2^{\Gamma+1}$. □

We next analyze an approximate bisection step.

Lemma 5. *Let f be a polynomial as in (1.1), $I = (a,b) \subset (-2^{\Gamma+2}, 2^{\Gamma+2})$ be an isolating interval for a root $\xi = z_{i_0}$ of f and $s = \operatorname{sign}(f(a))$. Then, Algorithm 2 applied on (f,I,s) requires a maximal precision of*

$$\begin{aligned}\rho_0 &:= 2\log(b-a)^{-1} + 4\log(d+1) + 8d + 10 + 2(d+1)\Gamma + \tau + 2\Sigma_f \\ &= O(\log(b-a)^{-1} + \tau + d\Gamma + \Sigma_f),\end{aligned}$$

and its bit complexity is bounded by $\tilde{O}(d(\log(b-a)^{-1} + \tau + d\Gamma + \Sigma_f))$.

Proof. Consider the three subdivision points $m_j := a + j \cdot \frac{b-a}{4}$, where $1 \leq j \leq 3$, and an arbitrary real root $z_i \neq \xi$ of f. Note that $|m_j - z_i| > \frac{b-a}{4}$ because the segment from m_j to z_i spans at least over a quarter of (a,b). Moreover, $|\xi - m_j| \leq \frac{3}{4}(b-a)$, and so

$$\sigma_i \leq |\xi - z_i| \leq |\xi - m_j| + |m_j - z_i| \leq \frac{3}{4}(b-a) + |m_j - z_i| \leq 4|m_j - z_i|.$$

It follows that m_j has a distance to z_i of at least $\frac{\sigma_i}{4}$. Hence, we can apply Lemma 4 to each m_j, that is, we have $|f(m_j)| > |\xi - m_j| \cdot 2^{-(2d+\Gamma+\Sigma_f)}$. Since the signs of f at the endpoints of I are known, it suffices to compute the signs of f at two of the three subdivision points. For at least two of these points, the distance of m_j to ξ is at least $\frac{b-a}{8}$, thus, we have $|f(m_j)| > |b-a| \cdot 2^{-(2d+3+\Gamma+\Sigma_f)}$ for at least two points. Then, due to Lemma 3, we can use interval arithmetic with a precision ρ to compute these signs if ρ satisfies

$$2^{-\rho+2}(d+1)^2 2^{\tau+d(\Gamma+2)} \leq (b-a)\cdot 2^{-(2d+3+\Gamma+\Sigma_f)},$$

which is equivalent to $\rho \geq \frac{\rho_0}{2}$. Since we double the precision in each step, we will eventually succeed with a precision smaller than ρ_0. The bit complexity for an arithmetic operation with fixed precision ρ is $\tilde{O}(\rho + d\tau)$. Namely, since the absolute value of each subdivision point is bounded by $O(\tau)$, the results in the intermediate steps have magnitude $O(d\tau)$ and we consider ρ bits after the binary point. At each subdivision point, we have to perform $O(d)$ arithmetic operations for the computation of $f(m_j)$, thus, the costs for these evaluations are bounded by $\tilde{O}(d(d\tau+\rho))$ bit operations. Since we double the precision in each iteration, the total costs are dominated by the last successful evaluation and, thus, we have to perform $\tilde{O}(d(\rho_0+d\tau)) = \tilde{O}(d(\log(b-a)^{-1}+d\tau+\Sigma_f))$ bit operations. □

We proceed with the analysis of an AQIR step. In order to bound the required precision, we need additional properties of the isolating interval.

Definition 6. *Let f be as in (1.1), $I := (a,b)$ be an isolating interval of a root ξ of f. We call I* normal[1] *if*

- $I \subseteq (-2^{\Gamma+2}, 2^{\Gamma+2})$,
- $|p - z_i| > \frac{\sigma_i}{4}$ *for every* $p \in I$ *and* $z_i \neq \xi$, *and*
- $\min\{|f(a)|, |f(b)|\} \geq 2^{-(28+2\tau+17d\Gamma+2\Sigma_f-5\log(b-a))}$.

In simple words, a normal isolating interval has a reasonable distance to any other root of f, and the function value at the endpoints is reasonably large. We will later see that it is possible to get normal intervals by a sequence of approximate bisection steps.

Lemma 7. *Let f be a polynomial as in (1.1), $I = (a,b)$ be a normal isolating interval for a root $\xi = z_{i_0}$ of f with $s = \operatorname{sign}(f(a))$, and let $N \leq 2^{2(\Gamma+4-\log(b-a))}$. Then, the* AQIR *step for (f,I,N,s) requires a precision of at most*

$$\rho_{max} := 87d\tau + 17d\Gamma + 4\Sigma_f - 14\log(b-a)$$

and, therefore, its bit complexity is bounded by

$$\tilde{O}(d(\tau + d\Gamma + \Sigma_f - \log(b-a))).$$

Moreover, the returned interval is again normal.

Proof. We have to distinguish two cases. For $N > 2$, we consider the two while-loops in Algorithm 3. In the first loop (Step 5), we evaluate $N\frac{f(a)}{f(a)-f(b)}$ via interval arithmetic,

[1] The reader may notice that the definition of "normal" depends on the upper bound Γ on Γ_f. Throughout our argument, we assume that such an initial Γ is given. We will finally choose a Γ which approximates Γ_f up to an (addative) error of $O(\log d)$.

doubling the precision ρ until the width of the resulting interval J is less than or equal to $1/4$. The following considerations show that we can achieve this if ρ fulfills

$$2^{-\rho+2}(d+1)^2 2^{\tau+d(\Gamma+2)} \leq \frac{\min(|f(a)|,|f(b)|)}{32N}. \tag{4.3}$$

W.l.o.g., we assume $f(a)>0$. If ρ fulfills the above condition, then, due to Lemma 3, $\mathfrak{B}(N\cdot f(a),\rho)$ is contained within the interval

$$[Nf(a)-\frac{|f(a)|}{32},Nf(a)+\frac{|f(a)|}{32}] = Nf(a)\cdot[1-\frac{1}{32N},1+\frac{1}{32N}]$$

and $\mathfrak{B}(f(a)-f(b),\rho)$ is contained within the interval

$$[f(a)-f(b)-\frac{|f(a)-f(b)|}{32N},f(a)-f(b)+\frac{|f(a)-f(b)|}{32N}] = (f(a)-f(b))\cdot[1-\frac{1}{32N},1+\frac{1}{32N}],$$

where the latter result uses the fact that $f(a)$ and $f(b)$ have different signs. It follows that $\mathfrak{B}(N\frac{f(a)}{f(a)-f(b)},\rho)$ is contained within $\frac{Nf(a)}{f(a)-f(b)}\cdot[(1-\frac{1}{32N})/(1+\frac{1}{32N}),(1+\frac{1}{32N})/(1-\frac{1}{32N})]$, and a simple computation shows that $N\cdot[(1-\frac{1}{32N})/(1+\frac{1}{32N}),(1+\frac{1}{32N})/(1-\frac{1}{32N})]$ has width less than $1/4$. Hence, since $\frac{f(a)}{f(a)-f(b)}$ has absolute value less than 1, $\mathfrak{B}(N\frac{f(a)}{f(a)-f(b)},\rho)$ has width less than $1/4$ as well. The bound (4.3) on ρ also writes as

$$\rho \geq 7+2\log(d+1)+\tau+d\Gamma+2d+\log N+\log\min(|f(a),f(b)|)^{-1}$$

and since we double ρ in each iteration, computing $N\frac{f(a)}{f(a)-f(b)}$ via interval arithmetic up to an error of $1/4$ demands for a precision

$$\begin{aligned}\rho &< 14+4\log(d+1)+2\tau+2d\Gamma+4d+2\log N+2\log\min(|f(a),f(b)|)^{-1}\\ &< 14+2\tau+10d\Gamma+2\log N+2\log\min(|f(a),f(b)|)^{-1},\end{aligned}$$

Since I is normal and because of the posed condition on N, we can bound this by

$$\begin{aligned}\rho &< 11d\tau+4(\tau+5-\log(b-a))+2(32d\tau+2\Sigma_f-5\log(b-a))\\ &< 87d\tau+4\Sigma_f-14\log(b-a)<\rho_{max}.\end{aligned}$$

We turn to the second while loop of Algorithm 3 (Steps 11-15) where f is evaluated at the subdivision points $m^*-\omega,m^*-\frac{7\omega}{8},\ldots,m^*+\omega$ as defined in (3.1). Since the interval is normal, we can apply Lemma 4 to each of the seven subdivision points. Furthermore, at least six of these points have distance $\geq\frac{b-a}{16N}$ to the root ξ and, thus, for these points, $|f|$ is larger than $\frac{b-a}{16N}\cdot 2^{-(2d+\tau+\Sigma_f)}$. Then, according to Lemma 4.3, it suffices to use a precision ρ that fulfills

$$2^{-\rho+2}(d+1)^2 2^{\tau+d(\Gamma+2)} \leq \frac{b-a}{16N}\cdot 2^{-(2d+\Gamma+\Sigma_f)}, \text{ or}$$

$$\rho \geq \rho_1 := 6+2\log(d+1)+\tau+d\Gamma+4d+\Gamma+\Sigma_f+\log N-\log(b-a).$$

The same argumentation as above then shows that the point evaluation will be performed with a maximal precision of less than

$$\begin{aligned} 2\rho_1 &< 2(6+\tau+7d\Gamma+\Gamma+\Sigma_f+\log N-\log(b-a)) \\ &\le 12+2\tau+14d\Gamma+2\Gamma+2\Sigma_f+4(\Gamma+4-\log(b-a))-\log(b-a) \\ &\le 28+2\tau+17d\Gamma+2\Sigma_f-5\log(b-a) \end{aligned}$$

which is bounded by ρ_{max}. Moreover, at the new endpoints a' and b', $|f|$ is at least

$$2^{-2\rho_1} \ge 2^{-(28+2\tau+17d\Gamma+2\Sigma_f-5\log(b-a))} \ge 2^{-(28+2\tau+17d\Gamma+2\Sigma_f-5\log(b'-a'))}$$

which proves that $I' = (a', b')$ is again normal.

It remains the case of $N = 2$, where a bisection step is performed. It is straightforward to see with Lemma 5 that the required precision is bounded by ρ_{max}, and in an analogue way as for the point evaluations for $N > 2$, we can see that the resulting interval is again normal. By the same argument as in Lemma 5, the overall bit complexity of the AQIR step is bounded by

$$\tilde{O}(d\rho_{max}) = \tilde{O}(d(d\tau+\Sigma_f-\log(b-a))). \qquad \square$$

5 Root refinement

We next analyze the complexity of our original problem: Given a polynomial f as in (1.1) and isolating intervals for all its real roots, refine the intervals to a size of at most 2^{-L}. Our refinement method consists of two steps. First, we turn the isolating intervals into normal intervals by applying bisections repeatedly. Second, we call the AQIR method repeatedly on the intervals until each has a width of at most 2^{-L}. Algorithm 5 summarizes our method for root refinement. We remark that depending on the properties of the root isolator used to get initial isolating intervals, the normalization can be skipped; this is for instance the case when using the isolator from [17]. We also emphasize that the normalization is unnecessary for the correctness of the algorithm; its purpose is to prevent the working precision in a single AQIR step of growing too high.

5.1 Normalization

The normalization (Algorithm 4) consists of two steps: first, the isolating intervals are refined using approximate bisection until the distance between two consecutive intervals is at least three times larger than the size of the larger of the two involved intervals. This ensures that all points in an isolating interval are reasonably far away from any other root of f. In the second step, each interval is enlarged on both sides by an interval of at least the same size as itself. This ensures that the endpoints are sufficiently far away from any root of f to prove a lower bound of f at the endpoints. W.l.o.g., we assume that the input intervals are contained in $(-2^{\Gamma+1}, 2^{\Gamma+1})$ because all roots are contained in that interval, so the leftmost and rightmost intervals can just be cut if necessary. Obviously, the resulting intervals are still isolating and disjoint from each other. Moreover, they do not become too small during the bisection process:

Algorithm 4 Normalization

INPUT: $f \in \mathbb{R}[t]$ a polynomial as in (1.1), $I_1 = (a_1, b_1), \ldots, I_m = (a_m, b_m)$ disjoint isolating intervals in ascending order, $s_1, \ldots, s_m$ with $s_k = \operatorname{sign}(f(\min I_k))$
OUTPUT: normal isolating intervals $J_1, \ldots, J_m$ with $z_k \in I_k \cap J_k$

```
procedure NORMALIZE($f, I_1, \ldots, I_m$)
    for k=1,...,m-1 do
        while $\min I_{k+1} - \max I_k < 3\max\{w(I_k), w(I_{k+1})\}$ do
            if $w(I_k) > w(I_{k+1})$
            then APPROXIMATE_BISECTION($f, I_k, s_k$)
            else APPROXIMATE_BISECTION($f, I_{k+1}, s_{k+1}$)
        end while
    end for
    for k=1,...,m-1 do
        $d_k \leftarrow \min I_{k+1} - \max I_k$
        $J_k \leftarrow [a_k - d_{k-1}/3, b_k + d_k/3]$ ▷ enlarge $I_k$ by more than $w(I_k)$ at both sides
    end for
    return $J_1, \ldots, J_m$
end procedure
```

Lemma 8. *For $J_1, \ldots, J_m$ as returned by Alg. 4, $w(J_k) \geq \frac{1}{3}\sigma_k$.*

Proof. After the first for-loop, the distance d_k between any two consecutive intervals I_k and I_{k+1} fulfills $d_k \geq 3\max\{w(I_k), w(I_{k+1})\}$, thus $\sigma_k < w(I_k) + w(I_{k+1}) + d_k < 2d_k$. Hence, in the last step, each I_k is enlarged by at least $\sigma_k/6$ on each side. This proves that the corresponding enlarged intervals J_k have size $\sigma_k/3$ or more. □

Lemma 9. *Algorithm 4 is correct, i.e., returns normal intervals.*

Proof. Let $J_1, \ldots, J_m$ denote the returned intervals, and fix some interval J_k containing the root z_k of f. We have to prove the three properties of Definition 6. The first property is clear because the initial interval are assumed to lie in $(-2^{\Gamma+1}, 2^{\Gamma+1})$. In the proof of Lemma 8, we have already shown that I_k is eventually enlarged by at least $\sigma_k/6$ on each side. More precisely, the right endpoint of J_k has distance at least $d_k/3 > \sigma_{k+1}/6$ to J_{k+1}, and the left endpoint of J_k has distance at least $d_{k-1}/3 > \sigma_{k-1}/6$ to J_{k-1}. It follows that, for each $x_0 \in J_k$, we have $|x_0 - z_{k\pm1}| \leq \sigma_{k\pm1}/3$, respectively. Hence, the second property in Definition 6 is fulfilled.

For the third property of Definition 6, let e be one of the endpoints of J_k. We have just proved that the distance to every root z_i except z_k is at least $\frac{\sigma_i}{3}$ and $|e - z_k| \geq \sigma_k/6$. With an estimation similar as in the proof of Lemma 4, we obtain:

$$|f(e)| \geq \frac{\sigma_k}{6} \prod_{i \neq k} \frac{\sigma_i}{3} \geq \frac{1}{8} \cdot \frac{1}{4^{d-1}} 2^{-\Sigma_f} = 2^{-(2d+\Sigma_f+1)},$$

and $2^{-(2d+\Sigma_f+1)} \geq 2^{-(28+2\tau+17d\Gamma+2\Sigma_f-5\log(b-a))}$ because $\log(b-a) \leq \Gamma+2$ and $-\Sigma_f \leq d(\Gamma+1) < 2d\Gamma$. □

Algorithm 5 Root Refinement

INPUT: $f = \sum a_i x^i \in \mathbb{R}[t]$ a polynomial as in (1.1), isolating intervals $I_1, \ldots, I_m$ for the real roots of f in ascending order, $L \in \mathbb{Z}$
OUTPUT: isolating intervals $J_1, \ldots, J_m$ with $w(J_k) \leq 2^{-L}$

```
procedure ROOT_REFINEMENT($f, L, I_1, \ldots, I_m$)
    $s_k := \mathrm{sign}(a_d) \cdot (-1)^{m-k+1}$                    ▷ $s_k = \mathrm{sign}(f(\min I_k))$
    $J_1, \ldots, J_m \leftarrow$ NORMALIZE($f, I_1, \ldots, I_m$)
    for k=1,...,m do
        $N \leftarrow 4$
        while $w(J_k) > 2^{-L}$ do $(J_k, N) \leftarrow$AQIR($f, J_k, N, s_k$)
    end for
    return $J_1, \ldots, J_m$
end procedure
```

Lemma 10. *Algorithm 4 has a complexity of*

$$\tilde{O}(d(d\Gamma + \Sigma_f)(\tau + d\Gamma + \Sigma_f))$$

Proof. As a direct consequence of Lemma 8, each interval I_k is only bisected $O(\Gamma + \log(\sigma_k)^{-1})$ many times because each starting interval is assumed to be contained in $(-2^{\Gamma+1}, 2^{\Gamma+1})$. So the total number of bisections adds up to $O(d\Gamma + \Sigma_f)$ considering all roots of f. Also, the size of the isolating interval I_k is lower bounded by $\frac{3}{20} \cdot \sigma_k = 2^{-O(\Sigma_f + d\Gamma)}$, so that one approximate bisection step has a complexity of $\tilde{O}(d(\tau + d\Gamma + \Sigma_f))$ due to Lemma 5. □

5.2 The AQIR sequence

It remains to bound the cost of the calls of AQIR. We mostly follow the argumentation from [11], mostly referring to that article for technical proofs. We introduce the following convenient notation:

Definition 11. *Let $I_0 := I$ be a normal isolating interval for some real root ξ of f, $N_0 := 4$ and $s := \mathrm{sign}(\min I_0)$. The* AQIR sequence $(S_0, S_1, \ldots, S_{v_\xi})$ *is defined by*

$$S_0 := (I_0, N_0) = (I, 4) \quad S_i = (I_i, N_i) := \text{AQIR}(f, I_{i-1}, N_{i-1}, s) \textit{ for } i \geq 1,$$

where v_ξ is the first index such that the interval I_{v_ξ} has width at most 2^{-L}. We say that $S_i \overset{\text{AQIR}}{\to} S_{i+1}$ succeeds *if* AQIR*(f, I_i, N_i, s) succeeds, and that $S_i \overset{\text{AQIR}}{\to} S_{i+1}$* fails *otherwise.*

As in [11], we divide the QIR sequence into two parts according to the following definition.

Definition 12. *For ξ a root of f, we define*

$$C_\xi := \frac{|f'(\xi)|}{8\left(\frac{d^2}{\sigma(\xi,f)}|f'(\xi)| + \sum_{i=2}^{d}\left(\frac{\sigma(\xi,f)}{d^2}\right)^{i-2}|f^{(i)}(\xi)|\right)}.$$

For $(S_0,\dots,S_{v_\xi})$ *the QIR sequence of* ξ*, define* k *as the minimal index such that* $S_k = (I_k,N_k) \overset{\text{AQIR}}{\to} S_{k+1}$ *succeeds and* $w(I_k) \leq C_\xi$*. We call* $(S_0,\dots,S_k)$ linear sequence *and* $(S_k,\dots,S_{v_\xi})$ quadratic sequence *of* ξ

Note that [11] defined a different threshold for splitting the qir sequence, and the linear sequence was called *initial sequence* therein. We renamed it to avoid confusion with the initial normalization phase in our variant.

Quadratic convergence. We start by justifying the name "quadratic sequence". Indeed, it turns out that all but one AQIR step in the quadratic sequence are successful, hence, N is squared in (almost) every step and therefore, the refinement factor of the interval is doubled in (almost) every step. We first prove two important properties of C_ξ as defined in Defition 12:

Lemma 13. *Let* $\xi \in \mathbb{C}$ *be a root of* f*.*

1. $0 < C_\xi \leq \frac{\sigma(\xi,f)}{8d^2}$
2. *Let* $\mu \in \mathbb{C}$ *be such that* $|\xi - \mu| < C_\xi$*. Then*

$$C_\xi < \frac{|f'(\xi)|}{8|f''(\mu)|}.$$

Proof. Note that all summands in the denominator of C_ξ are non-negative. Therefore, the first property follows immediately by removing all but the first summand in the denominator.

For the second property, we consider the Taylor expansion of $f''(\mu)$ in ξ:

$$f''(\mu) = \sum_{i=2}^{d} (\mu - \xi)^{i-2} \frac{f^{(i)}(\xi)}{(i-2)!}.$$

Because $|\mu - \xi| < C_\xi < \frac{\sigma(\xi)}{d^2}$ by the first property, we can bound

$$|f''(\mu)| < \sum_{i=2}^{d} \left(\frac{\sigma(\xi)}{d^2}\right)^{i-2} |f^{(i)}(\xi)|.$$

It follows that

$$\frac{|f'(\xi)|}{8|f''(\mu)|} > \frac{|f'(\xi)|}{8\left(\sum_{i=2}^{d} \left(\frac{\sigma(\xi)}{d^2}\right)^{i-2} |f^{(i)}(\xi)|\right)} > C_\xi$$

□

The following bound follows from considering the Taylor expansion of f at ξ in the expression for m:

Lemma 14. **[11, Thm. 4.8]** *Let (a,b) be isolating for ξ with width $\delta < C_\xi$ and m as in Lemma 2 (i.e., $m = a + \frac{f(a)}{f(a)-f(b)}(b-a)$). Then, $|m-\xi| \leq \frac{\delta^2}{8C_\xi}$.*

Proof. We consider the Taylor expansion of f at ξ. For a given $x \in (a,b)$, we have

$$f(x) = f'(\xi)(x-\xi) + \frac{1}{2}f''(\tilde{\xi})(x-\xi)^2$$

with some $\tilde{\xi} \in [x,\xi]$ or $\tilde{\xi} \in [\xi,x]$. Thus, we can simplify

$$\begin{aligned} |m-\xi| &= \left|\frac{f(b)(a-\xi)-f(a)(b-\xi)}{f(b)-f(a)}\right| = \left|\frac{\frac{1}{2}(f''(\tilde{\xi}_1)(b-\xi)^2(a-\xi)-f''(\tilde{\xi}_2)(a-\xi)^2(b-\xi))}{f(b)-f(a)}\right| \\ &\leq \frac{1}{2}|b-\xi||a-\xi| \cdot \frac{|f''(\tilde{\xi}_1)|(b-\xi)+|f''(\tilde{\xi}_2)|(\xi-a)}{|f(b)-f(a)|} \leq \frac{\delta^2 \max\{|f''(\tilde{\xi}_1)|,|f''(\tilde{\xi}_2)|\}}{2|f'(\nu)|} \end{aligned}$$

for some $\nu \in (a,b)$. The Taylor expansion of f' yields $f'(\nu) = f'(\xi) + f''(\tilde{\nu})(\nu-\xi)$ with $\tilde{\nu} \in (a,b)$. Since $\delta \leq C_\xi$, it follows with Lemma 13

$$|f''(\tilde{\nu})(\nu-\xi)| \leq |f''(\tilde{\nu})|C_\xi \leq \frac{1}{8}|f'(\xi)|.$$

Therefore $|f'(\nu)| > \frac{7}{8}|f'(\xi)| > \frac{1}{2}|f'(\xi)|$, and it follows again with Lemma 13 that

$$|m-\xi| \leq \frac{\delta^2 \max\{|f''(\tilde{\xi}_1)|,|f''(\tilde{\xi}_2)|\}}{|f'(\xi)|} \leq \frac{\delta^2}{8\frac{|f'(\xi)|}{8\max\{|f''(\tilde{\xi}_1)|,|f''(\tilde{\xi}_2)|\}}} < \frac{\delta^2}{8C_\xi}\square$$

Corollary 15. *Let I_j be an isolating interval for ξ of width $\delta_j \leq \frac{C_\xi}{N_j}$. Then, each call of the* AQIR *sequence*

$$(I_j,N_j) \overset{\text{AQIR}}{\to} (I_{j+1},N_{j+1}) \overset{\text{AQIR}}{\to} \ldots$$

succeeds.

Proof. We use induction on i. Assume that the first i AQIR calls succeed. Then, another simple induction shows that $\delta_{j+i} := w(I_{j+i}) \leq \frac{N_j\delta_j}{N_{j+i}} < \frac{C_\xi}{N_{j+i}}$, where we use that $N_{j+i} = N_{j+i-1}^2$. Then, according to Lemma 14, we have that

$$|m-\xi| \leq \delta_{j+i}^2\frac{1}{8C_\xi} \leq \delta_{j+i}\frac{C_\xi}{N_{j+i}}\frac{1}{8C_\xi} = \frac{1}{8}\frac{\delta_{j+i}}{N_{j+i}},$$

with m as above. By Lemma 2, the AQIR call succeeds. □

Corollary 16. **[11, Cor. 4.10]** *In the quadratic sequence, there is at most one failing* AQIR *call.*

Proof. Let $(I_i, N_i) \overset{\text{AQIR}}{\to} (I_{i+1}, N_{i+1})$ be the first failing AQIR call in the quadratic sequence. Since the quadratic sequence starts with a successful AQIR call, the predecessor $(I_{i-1}, N_{i-1}) \overset{\text{AQIR}}{\to} (I_i, N_i)$ is also part of quadratic sequence, and succeeds. Thus we have the sequence

$$(I_{i-1}, N_{i-1}) \overset{\substack{\textit{Sucess}\\ \text{AQIR}}}{\to} (I_i, N_i) \overset{\substack{\textit{Fail}\\ \text{AQIR}}}{\to} (I_{i+1}, N_{i+1})$$

One observes easily that $w(I_{i+1}) = w(I_i) = \frac{w(I_{i-1})}{N_{i-1}} \le \frac{C_\alpha}{N_{i-1}}$, and $N_{i+1} = \sqrt{N_i} = \sqrt{N_{i-1}^2} = N_{i-1}$. By Corollary 15, all further AQIR calls succeed. □

Cost of the linear sequence. We bound the costs of refining the isolating interval of ξ to size C_ξ with AQIR. We first show that, on average, the AQIR sequence refines by a factor two in every second step. This shows in particular that refining using AQIR is at most a factor of two worse than refining using approximate bisection.

Lemma 17. *Let $(S_0, \ldots, S_\ell)$ denote an arbitrary prefix of the* AQIR *sequence for ξ, starting with the isolating interval I_0 of width δ. Then, the width of I_ℓ is not larger than $\delta 2^{-(\ell-1)/2}$.*

Proof. Consider a subsequence $(S_i, \ldots, S_{i+j})$ of $(S_0, \ldots, S_\ell)$ such that $S_i \overset{\text{AQIR}}{\to} S_{i+1}$ is successful, but any other step in the subsequence fails. Because there are j steps in total, and thus $j-1$ consecutive failing steps, the successful step must have used a N with $N \ge 2^{2^{j-1}}$. Because $2^{j-1} \ge \frac{j}{2}$, it holds that

$$w(I_{i+j}) \le \frac{w(I_i)}{N} \le w(I_{i+j})2^{-2^{j-1}} \le w(I_{i+j})2^{-j/2}.$$

Repeating the argument for maximal subsequences of this form, we get that either $w(I_\ell) \le w(I_0)2^{-\ell/2}$ if the sequence starts with a successful step, or $w(I_\ell) \le w(I_0)2^{-(\ell-1)/2}$ otherwise, because the second step must be successful in this case. □

We want to apply Lemma 7 to bound the bit complexity of a single AQIR step. The following lemma shows that the condition on N from Lemma 7 is always met in the AQIR sequence.

Lemma 18. *Let $(I_j, N_j) \overset{\text{AQIR}}{\to} (I_{j+1}, N_{j+1})$ be a call in an* AQIR *sequence and $I_j := (a,b)$. Then, $N_j \le 2^{2(\Gamma+4-\log(b-a))}$.*

Proof. We do induction on j. Note that $I_0 \subset (-2^{\Gamma+2}, 2^{\Gamma+2})$ by normality, hence $b-a \le 2^{\Gamma+3}$. It follows that $2^{2(\Gamma+4-\log(b-a))} \ge 4 = N_0$. Assume that the statement is true for $j-1$. If the previous step $(I_{j-1}, N_{j-1}) \overset{\text{AQIR}}{\to} (I_j, N_j)$ is failing, then $N_j = \sqrt{N_{j-1}}$ and the isolating interval remains unchanged, so the statement is trivially correct. If the step is successful, then it holds that $(b-a) \le \frac{2^{\Gamma+3}}{\sqrt{N_j}}$. By rearranging terms, we get that $N_j \le 2^{2(\Gamma+3-\log(b-a))}$. □

It follows inductively that the conditions of Lemma 7 are met for each call in the AQIR sequence because I_0 is normal by construction. Therefore, the linear sequence for a root ξ of f is computed with a bit complexity of

$$\tilde{O}((\Gamma+\log(C_\xi)^{-1})d(\log(C_\xi^{-1})+\tau+d\Gamma+\Sigma_f)) \tag{5.1}$$

because $O(\Gamma+\log(C_\xi^{-1}))$ steps are necessary to refine the interval to a size smaller than C_ξ by Lemma 17, and the bit complexity is bounded by $\tilde{O}(d(\log(C_\xi^{-1})+\tau+d\Gamma+\Sigma_f))$ with Lemma 7. It remains to bound $\log(C_\xi)^{-1}$; we do so by bounding the sum of all $\log(C_\xi)^{-1}$ with the following lemma.

Lemma 19. $\sum_{i=1}^m \log(C_{z_i})^{-1} = O(d(\Gamma+\log d)+\Sigma_f))$

Proof. We note that

$$\sum_{\ell=1}^m \log(C_{z_\ell})^{-1} = \sum_{\ell=1}^m \log\left(8\cdot\left(\frac{d^2}{\sigma_\ell}+\sum_{i=2}^d\left(\frac{\sigma_\ell}{d^2}\right)^{i-2}\left|\frac{f^{(i)}(z_\ell)}{f'(z_\ell)}\right|\right)\right).$$

We focus on the quotient $\left|\frac{f^{(i)}(z_\ell)}{f'(z_\ell)}\right|$. Let $z'_1,\dots,z'_{d-1}$ denote the (not necessarily distinct) roots of f'. Note that for $x\in\mathbb{C}$ and any $i\geq 1$,

$$f^{(i)}(x) = a_d \sum_{\substack{X\subseteq\{1,\dots,n-1\}\\ |X|=i-1}} \prod_{\substack{j\in\{1,\dots,d-1\}\\ j\notin X}} (x-z'_j)$$

Therefore, the quotient writes as

$$\left|\frac{f^{(i)}(z_\ell)}{f'(z_\ell)}\right| = \left|\sum_{\substack{X\subseteq\{1,\dots,d-1\}\\ |X|=i-1}} \prod_{j\in X}\frac{1}{z_\ell-z'_j}\right| \leq \sum_{\substack{X\subseteq\{1,\dots,d-1\}\\ |X|=i-1}} \prod_{j\in X}\frac{1}{|z_\ell-z'_j|}.$$

Since $|z_\ell-z'_j|\geq\frac{\sigma_\ell}{d}$ [8, Thm.8], we can further bound this to

$$\sum_{\substack{X\subseteq\{1,\dots,d-1\}\\ |X|=i-1}} \prod_{j\in X}\frac{1}{|z_\ell-z'_j|} \leq \sum_{\substack{X\subseteq\{1,\dots,d-1\}\\ |X|=i-1}} \prod_{j\in X}\frac{d}{\sigma_\ell} \leq \sum_{\substack{X\subseteq\{1,\dots,d-1\}\\ |X|=i-1}}\left(\frac{d}{\sigma_\ell}\right)^{i-1} \leq d^{i-1}\left(\frac{d}{\sigma_\ell}\right)^{i-1} = \frac{d^{2i-2}}{\sigma_\ell^{i-1}},$$

and, therefore,

$$\sum_{i=2}^d\left(\frac{\sigma_\ell}{d^2}\right)^{i-2}\left|\frac{f^{(i)}(z_\ell)}{f'(z_\ell)}\right| \leq \sum_{i=2}^d\left(\frac{\sigma_\ell}{d^2}\right)^{i-2}\frac{d^{2i-2}}{\sigma_\ell^{i-1}} = \sum_{i=2}^d\frac{d^2}{\sigma_\ell} = (d-1)\frac{d^2}{\sigma_\ell}.$$

Plugging in into the overall sum yields

$$\begin{aligned}\sum_{\ell=1}^m \log(C_{z_\ell})^{-1} &= \sum_{\ell=1}^m \log\left(8\cdot\left(\frac{d^2}{\sigma_\ell}+(d-1)\frac{d^2}{\sigma_\ell}\right)\right) = 3d+\sum_{\ell=1}^m\log\frac{d^3}{\sigma_\ell}\\ &= 3d+3m\log d+\Sigma_f+\sum_{\ell=m+1}^d\log\sigma_\ell \leq 3d+3d\log d+\Sigma_f+d(\Gamma+1) = O(d(\Gamma+\log d)+\Sigma_f)\square\end{aligned}$$

Lemma 20. *The linear sequences for all real roots are computed within a total bit complexity of*

$$\tilde{O}(d(d\Gamma+\Sigma_f)(\tau+d\Gamma+\Sigma_f)$$

Proof. The total cost of all linear sequences is bounded by

$$\tilde{O}(\sum_{i=1}^{m}(\Gamma+\log(C_{z_i}^{-1}))d(\log(C_{z_i}^{-1})+\tau+d\Gamma+\Sigma_f)).$$

By rearranging terms, we obtain

$$= \tilde{O}(d^2\Gamma(\tau+d\Gamma+\Sigma_f)+d(\tau+d\Gamma+\Sigma_f)\sum\log(C_{z_i}^{-1})+d(\sum\log(C_{z_i}^{-1}))^2)$$

which equals $\tilde{O}(d(d\Gamma+\Sigma_f)(\tau+d\Gamma+\Sigma_f))$ with Lemma 19. □

Cost of the quadratic sequence. Let us fix some root ξ of f. Its quadratic sequence consists of at most $1+\log L$ steps, because N is squared in every step (except for at most one failing step) and the sequence stops as soon as the interval is smaller than 2^{-L}. Since we ignore logarithmic factors, it is enough to bound the costs of one QIR step in the sequence. Clearly, since the interval is not smaller than 2^{-L} in such a step, we have that $\log(b-a)^{-1} \leq L$. Therefore, the required precision is bounded by $O(L+\tau+d\Gamma+\Sigma_f)$. It follows that an AQIR step performs up to $\tilde{O}(d(L+\tau+d\Gamma+\Sigma_f))$ bit operations.

Lemma 21. *The quadratic sequences for one real root is computed within a bit complexity of*

$$\tilde{O}(d(L+\tau+d\Gamma+\Sigma_f)).$$

Total cost. We have everything together to prove the main result

Theorem 22. *Algorithm 5 performs root refinement within*

$$\tilde{O}(d(d\Gamma_f+\Sigma_f)^2+dL)$$

bit operations for a single real root [2] *of f, and within*

$$\tilde{O}(d(d\Gamma_f+\Sigma_f)^2+d^2L)$$

for all real roots. The coefficients of f need to be approximated to $\tilde{O}(L+d\Gamma_f+\Sigma_f)$ bits after the binary point.

Proof. We first restrict to the case where $1 \leq |a_d| < 2$. The so far achieved complexity bounds are formulated in terms of an arbitrary (but given) upper bound $\Gamma \in \mathbb{N}$ on Γ_f. In [17, Section 6.1], it is shown how to compute a Γ with $\Gamma_f \leq \Gamma < \Gamma_f + 4\log d$ using $\tilde{O}((d\Gamma_f)^2)$ bit operations and approximations of f to $\tilde{O}(d\Gamma_f)$ bits after the binary point. Furthermore, the latter construction also shows that $\tau = \lceil\log(\max_i|a_i|)\rceil =$

[2] In its initial formulation, Algorithm 5 assumes that isolating intervals for *all* real roots are given. If only one isolating interval I_k for a root z_k is given, we have to normalize I_k first and, then, compute the signs of f at the endpoints of I.

$O(d\Gamma)$ if $1 \leq |a_d| < 2$. By Lemma 10, the normalization for all isolating intervals requires $\tilde{O}(d(d\Gamma + \Sigma_f)(\tau + d\Gamma + \Sigma_f))$ bit operations. The linear subsequences of the AQIR sequence are computed in the same time by Lemma 20. The quadratic subsequences are computed with $\tilde{O}(d^2L + d^2\tau + d^3\Gamma + d^2\Sigma_f)$ bit operations by Lemma 21; the latter three terms are all dominated by $\tilde{O}(d(d\Gamma + \Sigma_f)(\tau + d\Gamma + \Sigma_f))$. Hence, with $\Gamma = O(\Gamma_f + \log d)$ as above and $\tau = \tilde{O}(d\Gamma_f)$, the claimed bound on the bit complexity to refine all roots follows. The maximal number of required bits follows from Lemma 7 because the maximal required precision in any AQIR step is bounded by $O(L + \tau + d\Gamma + \Sigma_f) = \tilde{O}(L + d\Gamma_f + \Sigma_f)$. The bound on refining a single root follows easily when considering the cost of the quadratic sequence for this root only.

For the more general case, where $1 \leq |a_d| < 2$ is not necessarily given, we first shift the coefficients by $s = \lfloor \log |a_d| \rfloor$ bits such that we can apply the above result to the shifted polynomial. Since this coefficient shift does not change the roots, our bit complexity bound follows immediately. For the required precision, we need $\tilde{O}(L + d\Gamma_f + \Sigma_f) - s$ since we need an approximation of the shifted polynomial to $\tilde{O}(L + d\Gamma_f + \Sigma_f)$ bits after the binary point. □

For integer polynomials, we have $\Gamma_f = O(\tau)$ and $\Sigma_f = \tilde{O}(d\tau)$ [17, §7.2]. Thus, it follows

Corollary 23. *If f is a polynomial with integer coefficients of maximal bitsize τ, the bit complexity of Algorithm 5 is bounded by*

$$\tilde{O}(d^3\tau^2 + d^2L).$$

This improves the bound from [11] by a factor of d.

6 Concluding Remarks

We have presented a complete solution to the root refinement problem using validated numerical methods in this paper. Despite the relative simplicity of the approach, we obtain a bit complexity which is essentially competitive to best known bounds which have been achieved by much more sophisticated algorithms.

We have shown that the complexity of approximating roots of a real polynomial only depends on the geometry of the roots and not on the complexity or the type of the coefficients. By means of the following example, we demonstrate how to benefit from such an approach. Assume that the root refinement problem is applied to a non-square-free integer polynomial F of degree d and coefficient size τ. It is known that its square-free part f can be computed in $\tilde{O}(d^3\tau)$ operations using the Euclidean algorithm and f is of degree at most d and has coefficient size at most $\tau + d$. However, the roots of F and f coincide, so it is possible to apply Cauchy's bound on F (instead of on f) which yields a root bound $\Gamma \in O(\tau)$ (in comparison to $\Gamma \in O(\tau + d)$ when Cauchy's Bound is applied to the square-free part). This finally leads to a complexity of $\tilde{O}(d^3\tau^2 + d^2L)$ for non-square-free polynomials as well.

Although the focus of this work was the asymptotic complexity, the presented algorithm also aims for a practically efficient solution of the root approximation problem.

Indeed, a simplified version of our approach (for integer coefficients) is included in the recently introduced CGAL[3]-package on algebraic computations [3]. Experimental comparisons in the context of [2] have shown that the approximate version of QIR gives significantly better running times than its exact counterpart. These observations underline the practical relevance of our approximate version and suggest a practical comparison with state-of-the-art solvers as further work.

References

[1] J. Abbott. Quadratic Interval Refinement for Real Roots. Poster presented at the *2006 Intern. Symp. on Symbolic and Algebraic Computation (ISSAC 2006)*, 2006.

[2] E. Berberich, P. Emeliyanenko, and M. Sagraloff. An elimination method for solving bivariate polynomial systems: Eliminating the usual drawbacks. In *Workshop on Algorithm Engineering & Experiments (ALENEX)*, pages 35–47, 2011.

[3] E. Berberich, M. Hemmer, and M. Kerber. A generic algebraic kernel for non-linear geometric applications. Research report 7274, INRIA, 2010.

[4] J. Bus and T.J.Dekker. Two efficient algorithms with guaranteed convergence for finding a zero of a function. *ACM Trans. on Math. Software*, 1(4):330–345, 1975.

[5] J. Cheng, S. Lazard, L. Pearanda, M. Pouget, F. Rouillier, and E. Tsigaridas. On the topology of real algebraic plane curves. *Mathematics in Computer Science*, 4:113–137, 2010.

[6] G. E. Collins and A. G. Akritas. Polynomial Real Root Isolation Using Descartes' Rule of Signs. In *Proc. of the 3rd ACM Symp. on Symbolic and Algebraic Computation (SYMSAC 1976)*, pages 272–275. ACM Press, 1976.

[7] Z. Du, V. Sharma, and C. Yap. Amortized bound for root isolation via Sturm sequences. In *Symbolic-Numeric Computation*, Trends in Mathematics, pages 113–129. Birkhuser Basel, 2007.

[8] A. Eigenwillig. On multiple roots in Descartes' rule and their distance to roots of higher derivatives. *Journal of Computational and Applied Mathematics*, 200(1):226–230, March 2007.

[9] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In *Proc. of the 2007 Intern. Symp. on Symbolic and Algebraic Computation (ISSAC 2007)*, pages 151–158, 2007.

[10] A. Eigenwillig, L. Kettner, W. Krandick, K. Mehlhorn, S. Schmitt, and N. Wolpert. A Descartes algorithm for polynomials with bit-stream coefficients. In *8th International Workshop on Computer Algebra in Scientific Computing (CASC 2005)*, volume 3718 of *LNCS*, pages 138–149, 2005.

[3] Computational Geometry Algorithms Library, `www.cgal.org`

[11] M. Kerber. On the complexity of reliable root approximation. In *11th International Workshop on Computer Algebra in Scientific Computing (CASC 2009)*, volume 5743 of *LNCS*, pages 155–167. Springer, 2009.

[12] M. Kerber and M. Sagraloff. Efficient real root approximation. In *Accepted for the 2011 International Symposium on Symbolic and Algebraic Computation (ISSAC 2011)*, 2011.

[13] K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *CGTA*, 2011. to appear; for a draft, see *http://www.mpi-inf.mpg.de/˜msagralo/cpgeneral.pdf.*

[14] V. Y. Pan. Optimal and nearly optimal algorithms for approximating polynomial zeros. *Computers and Mathematics with Applications*, 31(12):97–138, 1996.

[15] V. Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.

[16] F. Rouillier and P. Zimmermann. Efficient isolation of polynomial's real roots. *Journal of Compututational and Applied Mathematics*, 162(1):33–50, 2004.

[17] M. Sagraloff. On the complexity of real root isolation. arXiv:1011.0344v2, 2011.